Development of the Tumbleweed Rover

Jaret Matthews

Robotic Vehicles Group
NASA Jet Propulsion Laboratory
Pasadena, California
jaret.b.matthews@jpl.nasa.gov

May 2003

Acknowledgements

Thanks to the following people for their support throughout the internship period:

Dr. Alberto Behar, NASA Jet Propulsion Laboratory
Dr. John Farrow, International Space University
Brian Wilcox, NASA Jet Propulsion Laboratory

Note: The following document was originally generated as an Individual Project Report and submitted to the International Space University in partial fulfillment of the requirements of the Degree of Master of Space Studies.

Table of Contents

List of Figures

List of Tables

Abstract

This report documents the development of the Tumbleweed Rover at the NASA Jet Propulsion Laboratory in Pasadena, California, USA. The report covers the work done by the author on the rover hardware and software, the creation of a ground station and its accompanying software, as well as descriptions of recent and proposed future field tests of the rover.

1. Introduction

1.1. Background

The Tumbleweed rover, currently under development at the NASA Jet Propulsion Laboratory (JPL), has a rather serendipitous origin. While researchers in JPL's Inflatable Technology for Robotics Program were testing a three -wheeled, inflatable rover in the Mojave desert (see Figure 1.1), one of its 1.5 m diameter yellow tires was ripped off by a gust of wind. The runaway tire quickly picked up speed in the moderate wind and seemed unimpeded by the desert's rough terrain. The renegade ball was able to climb steep slopes, over large boulders, and through the jagged brush without hesitation. This seemingly unlucky incident produced a rather lucky discovery and was the inspiration for the Tumbleweed rover.

Figure 1.1 – JPL's Inflatable Rover

[NASA JPL]

The Tumbleweed rover derives its name from the dead sagebrush balls that blow across the deserts of the American southwest. Likewise, the rover's only means of locomotion is the ambient wind. A 6-meter diameter Tumbleweed is envisaged for deployment on the surface of Mars. Such a ball could potentially serve as its own descent and landing system, replacing parachutes and airbags. While its mobility is dependant on the wind, the Tumbleweed rover will have some ability to control its speed by regulating its level of internal pressure. By partially deflating itself, the ball will have the ability to slow or stop before re-inflating and continuing on with the next batch of wind [Jones 2001].

The Martian Tumbleweed would be comprised of a 20 kg ball and a 20 kg payload suspended from the center. Traveling at speeds up to 10 m/s in the 20 m/s wind of a typical Martian afternoon, the ball is expected to climb 20° slopes with ease. In stronger winds (30 m/s), the ball should be able to climb slopes as high as 45° [Jones 2001]. Even in the thin Martian atmosphere, the large balls should have the ability climb over one-meter rocks [JPL Press Release 2001]. It is estimated that this performance is sufficient enough to traverse some 99% of the Martian surface.

In the 1960s, JPL experimented with similar balls for use as landing systems. In impacts as high as 60 m/s, with payload fractions as high as 75%, the spheres were shown to survive. The Martian Tumbleweed by comparison, would only experience a terminal velocity of 30 m/s and have a payload fraction around 30% [Jones 2001]. The material that comprises the ball itself is Pathfinder heritage (landing airbags) and is also being used on the upcoming Mars Exploration Rovers [JPL Press Release 2001].

Because Tumbleweed does not require dedicated descent, landing, and/or mobility systems, it can have a large amount of its total mass reserved for science payloads. In addition, because it remains deflated en-route to Mars, it can achieve an extremely low-packing volume when compared to other, more traditional, rovers.

Once deployed on the surface the Tumbleweed rover would traverse vast distances while consuming minimal power. After reaching an area of scientific interest, it will partially deflate such that it comes to a stop (see Figure 1.2). The centrally mounted payload will then begin deploying instruments, and perhaps even a drill, onto the Martian surface. After colleting the required samples or readings, data would then be beamed up to an orbiting satellite for relay back to Earth. Following transmission of the data, Tumbleweed would re-inflate, await the next gust of wind, and move on to the subsequent areas of interest.

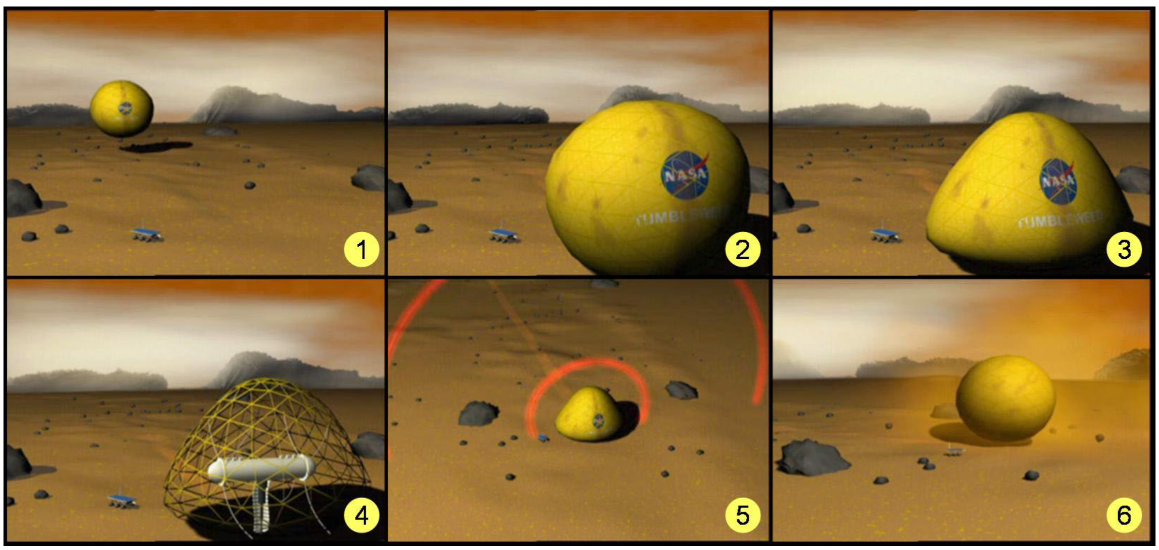


Figure 1.2 – (1) The Tumbleweed Rover is blown around on Mars. (2 – 3) Tumbleweed deflates to stop at an area of scientific interest. (4) Instruments and a drill are deployed to the surface to take measurements and samples. (5) Data is transmitted to an orbiting satellite which relays the data back to Earth (6) Tumbleweed re-inflates and continues on its journey.

Generally speaking, any such rover will be at the complete mercy of the wind. One of the more likely failure modes comes from being blown into a local depression, such that even ultra-fast winds would not be sufficient to dislodge the ball. One means of avoiding this scenario is by repositioning the center of mass while rolling, giving it some ability to steer. For example, an oblong (pill-shaped) Tumbleweed could pump fluid back and forth along the central axis to affect the direction of travel to some degree [JPL Press Release 2001].

1.2. Recent Developments

Mars might not be the only place where a rover like Tumbleweed is of use. The current round of development for Tumbleweed is focused on an Antarctic deployment in the 2004 field season. While this test will serve to evaluate Tumbleweed in thermal conditions similar to those seen on Mars (at mid latitude during the summer) and on terrain comparable to that found at the Martian poles, it will also make substantial contributions to Antarctic science. The type of data that can be collected by Tumbleweed in the Antarctic environment includes:

- Temperature
- Ice thickness via subsurface radar
- Topographic data via GPS
- Ground UV intensities (ozone depletion)
- Subsurface meteorite detection (via magnetometer)

At this stage, Tumbleweed is able to acquire temperature, pressure, accelerometer, and position data. This data is transmitted to a waiting ground station via the Iridium satellite network from anywhere in the world. If and when Tumbleweed is deployed in Antarctica it will be done without any intention to recover the hardware. If all goes well, the rover will traverse hundreds, if not thousands, of kilometers to the Antarctic coast from its release point at the South Pole.

In preparation for the Antarctic deployment, Tumbleweed has recently undergone several hardware and software upgrades, as well as a series of field tests to validate the new systems. In parallel with the development of the rover itself, ground station hardware and software have also been developed. The new ground station equipment and its associated software enables the acquisition of Tumbleweed data from anywhere in the world. If while operating, the new ground station happens to have access to an internet connection, it can also serve up the data live for remote viewing on the web.

The following document describes the Tumbleweed rover system as it exists now and serves as a record of its development from February to May of 2003.

2. Tumbleweed Systems Overview

2.1. System Architecture

The Tumbleweed system architecture is described in Figure 2.1. After acquiring temperature, pressure, accelerometer, and position information from anywhere in the world, Tumbleweed transmits its data to the Iridium satellite network (see (1) of Figure 2.1). Using intersatellite links, Iridium passes the data to the waiting ground station - also located anywhere in the world (see (2) of Figure 2.1). Next, the ground station receives, parses, saves, and displays the incoming data (see (3) of Figure 2.1). The main ground station also runs a web server. The web server ensures that the live incoming data is made available for anyone to see via the internet (see (4) of Figure 2.1).

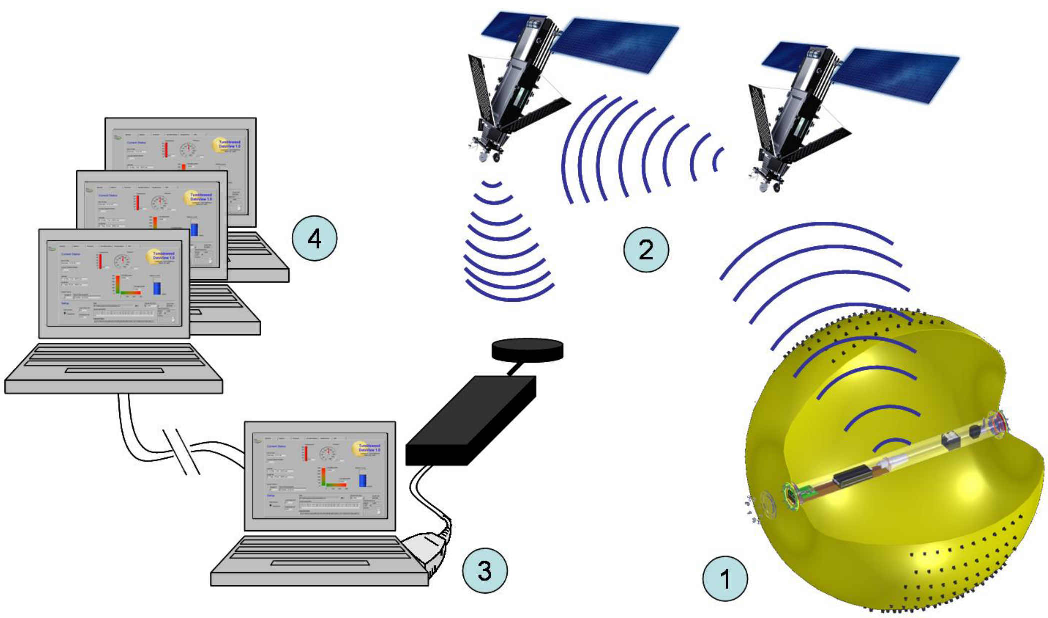


Figure 2.1 – The Tumbleweed System Architecture

2.2. Rover System Overview

The most recent configuration of the Tumbleweed rover can be seen in Figure 2.2. The body of the rover consists of a 5 foot diameter nylon bag with rubber studs applied to the outside. While the rubber studs were originally applied to boost traction on the 3-wheeled, inflatable rover (see introduction), the studs actually help Tumbleweed to roll along a preferred axis. Mounted inside the body of the rover along the preferred axis of rotation is a 4-foot long lexan tube. This tube is purposefully shorter than the diameter of the ball to prevent the ends of the tube from impacting the ground when Tumbleweed fails to roll along the intended axis.

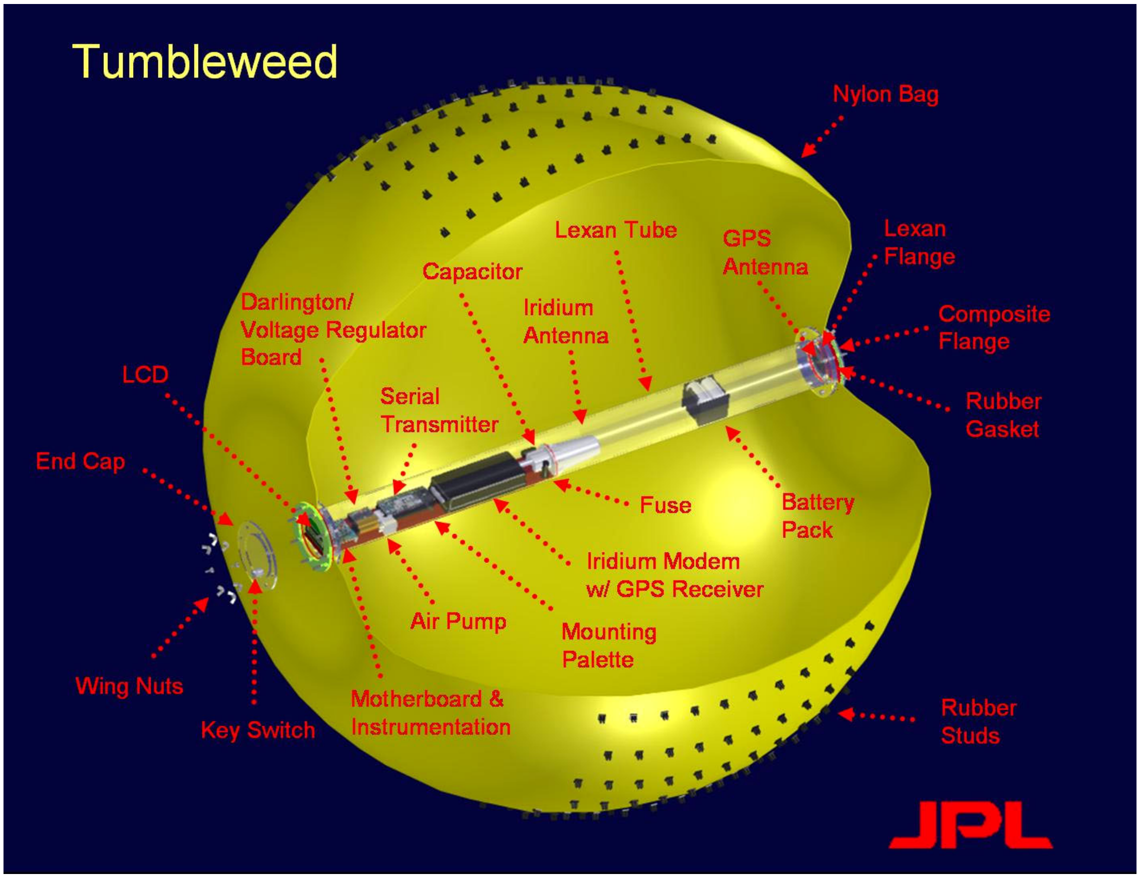


Figure 2.2 – Exploded view of the Tumbleweed rover.

Inside the lexan tube is housed the electronics package of the rover. This package consists of a motherboard, a liquid crystal display (LCD), a 900 MHz serial transmitter, an Iridium modem with integrated GPS receiver, an omni-directional Iridium antenna, an active GPS antenna, a lithium battery pack, a pulse modulated voltage regulator board, a darling transistor board, and an air pump. The central tube is attached to the nylon membrane on each side by a set of lexan and composite flanges and eight stainless steel bolts. Sandwiched between the lexan and composite flanges with the nylon bag are plyable rubber gaskets to prevent the leakage of air (for a detailed list of all of the Tumbleweed components, including their size and weight, see Table 2.1).

The motherboard consists of numerous components that serve to control Tumbleweed, as well as take scientific data. Mounted on the board are two pressure transducers (one for ambient pressure and the other for monitoring the membrane's internal pressure), a thermocouple (for recording ambient temperature), three 2-axis accelerometers (to determine the orientation of tumbleweed at the time of acquisition), and a real-time clock (for noting the time at which the readings were made).

The heart of the rover's electronics package is the Basic Stamp microcontroller, which is also mounted on the motherboard. The microcontroller has 15 I/O pins which are used to communicate with the various components and instruments. The microcontroller takes temperature, pressure, accelerometer, battery level, time, and position data once every second. This data is stored as a string of 8 bit numbers in the Stamp's 512K Flash EEPROM memory (e.g. S, 255, 255, 255, 255, etc). Every fifteen minutes, the microcontroller attempts to make a call with the Iridium modem. If a connection is made, the Stamp begins sending the stored strings of data. If the connection is unsuccessful, the Stamp will try once more. If the second attempt is unsuccessful, the Stamp will continue taking new data and wait another fifteen minutes before making the next call attempt.

Each time the Stamp acquires a new set of data (i.e. – every second) it transmits the information locally via the 900 MHz serial transmitter. This transmitter has about the same range as the average household cordless phone. This capability is extremely useful for the development phase, but will not likely be present on the final version. The serial transmitter also allows for the remote updating of the onboard software. Therefore, in the field it is possible to completely change the software that is running on the microprocessor without removing the electronics package. The same information that is sent to the serial transmitter is also sent to the LCD. Again, this capability helpful for the initial field testing, but is not necessary for the Antarctic deployment.

What really makes Tumbleweed unique is the Iridium modem. This component allows data to be sent from anywhere on the planet. Further, because Iridium is polar orbiting, the performance of the modem should improve with latitude, thus making it the ideal system for use in the Antarctic. The modem is unique in that it has an integrated GPS receiver. To make phone calls with the modem, the Stamp microcontroller sends serial commands written in the Hayes AT command set that is commonly used by most conventional modems. The GPS data is retrieved from the modem by similar commands. Just as with the other data types, the Stamp requests the GPS data once every second, parses, and saves it to the EEPROM.

The current Tumbleweed rover consists of the following components:

Quantity	Description	Dimensions	Mass (kg)
1	Motherboard		
	1 Stamp microcontroller → Serial EEPROM (32K) → Flash EEPROM (512K) → 15 I/O pins		
	1 Digital thermocouple		
	2 Analog pressure transducers		
	2 2-axis accelerometers		
	1 Real-time clock		
1	Iridium modem (2.4 Kb/s) with integrated GPS unit	3.15” x 26” x 3.15”	1.778 kg
1	Iridium omni-directional antenna		
1	Active GPS antenna		
1	Diaphragm air pump and associated tubing		
1	Darlington/Voltage Regulator board		
1	Liquid Crystal Display		
1	900 MHz Serial Transmitter		
1	In-line Fuse (4 A, Slow-burn)		
1	Composite Mounting Pallet		
1	Lithium Battery Pack (Small/Large) → 4 ‘D’ Cells/ 24 ‘D’ Cells → 7.2 V → 26 Ah / 156 Ah	Each battery: 2.25” x 1.25” dia Small pack: 2.75” x 2.75” x 2.5” Large pack: 2.75” x 16.5” x 2.5”	Each battery: 97.7 g Small pack: 58 g empty 447.8 g w/ batteries Large pack: 360 g empty 2.705 kg w/ batteries
	or NiMH Battery Pack (Small/Large) → 8 ‘D’ Cells/ 24 ‘D’ Cells → 9.6 V → 8 Ah/ 24 Ah	Each battery: 2.25” x 1.25” dia Small pack : 2.75” x 5.5” x 2.5” Large pack: 2.75” x 16.5” x 2.5”	Each battery: 169.7 g Small pack: 106 g empty 1.458 kg w/ batteries Large pack: 330 g empty 4.403 kg w/ batteries
1	Lexan central tube	4’ x 4”OD x 3.75” ID	1.392 kg
16	¼” x 20 SS allen bolts	1.25”	8.5 g each
16	¼” x 20 brass wing nuts	-	5.3 g each
1	Key operated on/off switch	1.5” x .75” dia	150 g
2	Lexan flanges	6”OD x 4”ID	141.4 g each
2	Lexan end plates	6”OD	178.3 g each
2	Rubber gaskets	6”OD x 4”ID	20.2 g each
2	Composite flanges	6”OD x 3.75” ID	136.7
1	Nylon ball	5’ dia sphere	2.918 kg
	Total	5’ dia sphere	7.860 kg w/ lightest batt pack 11.815 kg w/ heaviest batt pack

Table 2.1 – Tumbleweed subsystem size* & mass breakdown

** The sizes are given in imperial units to be most useful to a reader in the US interested in finding the described components*

3. Tumbleweed Hardware Development

3.1. Central Tube Replacement

The original Tumbleweed electronics package used an off-the-self handheld GPS receiver to determine its position and was small enough to fit into a 3-inch internal diameter central tube. However, in an effort to simplify the overall system by reducing the components, the next version was to include a new Iridium modem with an integrated GPS receiver. This new modem has a larger casing and therefore requires a larger diameter central tube. A 3.75-inch internal diameter tube was determined via CAD modeling to be the smallest diameter tube still able to accommodate the new modem. The material to construct this tube, its associated flanges, gaskets, and end caps was procured and the various components were manufactured in the JPL machine shop (see Figure 3.1). Because there is a chance that another Tumbleweed will be deployed in Greenland this summer, a second set of flanges, gaskets, and end caps was constructed.

Figure 3.1 – Manufacturing the larger diameter flanges in the machine shop.

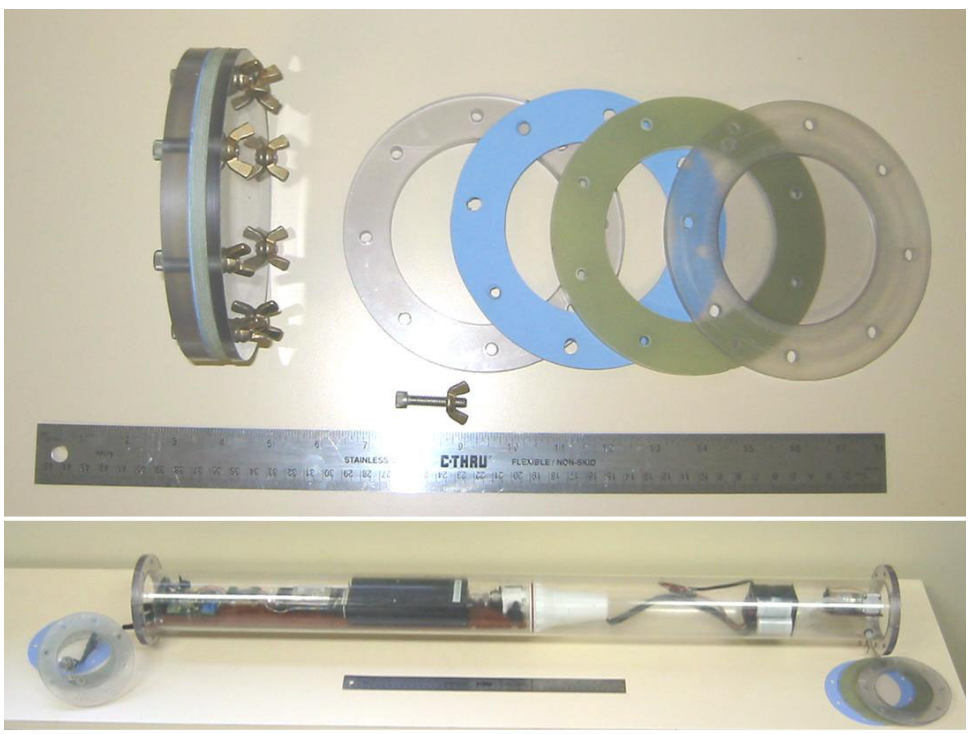


Figure 3.2 – Top: The new flanges, gaskets, and endcaps. Bottom: The electronics pallet and battery pack slide snuggly into the new central tube.

3.2. Electronics Pallet Design & Construction

The pallet on which most of Tumbleweed's components are mounted has undergone several design iterations. The primary function of this pallet is to suspend and secure the electronics within the central tube. The pallet also serves as an easy way to insert and remove the electronics without disconnecting many of the associated wires and cables. After several rounds of testing, the final pallet design was settled on (see Figure 3.3). This design places the omni-directional Iridium antenna exactly in the center of the inflated Tumbleweed. This placement ensures that the antenna will be at least 2.5 feet off of the ground at all times, regardless of orientation. This height was determined in earlier testing to be sufficient to maintain good signal strength while rolling. The final pallet is constructed of thin, lightweight, flexible, composite material and has performed well in several rounds of field testing.

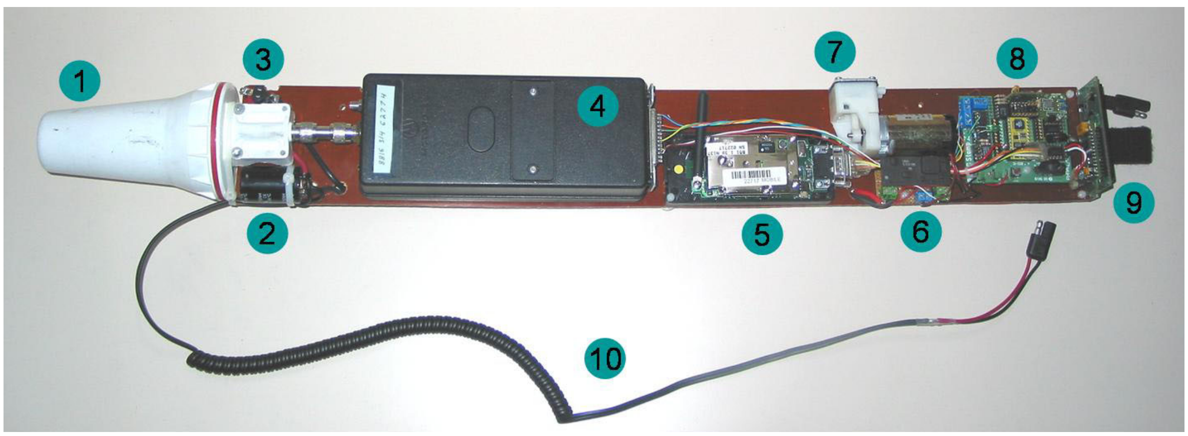


Figure 3.3 – The final electronics mounting pallet. (1) Iridium omni-directional antenna (2) Capacitor (3) Fuse (4) Iridium modem (5) Serial transmitter (6) Darlington/Voltage Regulator Board (7) Pump (8) Stamp microcontroller and various instrumentation (9) LCD screen and pull strap (10) Coiled battery cable.

3.3. Power System Design & Construction

By far, the most problem ridden area of Tumbleweed's development has been in the design and construction of the power system. The main difficulty rests in the fact that there are essentially six different power consuming systems on Tumbleweed, each requiring a unique power condition. The motherboard prefers 7.5 volts and resets if it does not receive at least 4.2 volts. The LCD is designed for 5 volts but operates (albeit poorly) down to about 4 volts and up to 6 volts. The Iridium modem prefers 4.4 volts and will reset at anything more than about 5.1 volts. The pump is designed for 12 volts, but will operate (only slower) down to about 6 volts. The serial transmitter can be operated at 7.5 volts via its normal power inlet or at 5 volts off of its serial pins, but it will not handle much deviation in either case. And finally, the real-time clock operates off of 3 volts.

An additional constraint that is placed on the power system is that it has to accommodate various power sources. For lab testing, Tumbleweed runs off of a 120 VAC to 12 VDC power supply. For preliminary field testing, it runs off of rechargeable nickel metal hydride battery packs producing 9.6VDC, while for the final deployment it will run off of a Lithium battery pack producing 7.2VDC. With so many systems requiring unique voltages to operate, it makes designing the power system quite difficult. This problem becomes even more difficult when the means of supplying power is in flux.

Again, after several iterations, the final design of the power system was settled on. A schematic of this system can be seen in Figure 3.4. To handle the various power sources, a pulse modulated voltage regulator is used to knock any voltage from 6 to 16 volts down to 5 volts. Rather than wasting the energy as heat, this regulator pulses the input voltage to achieve an average of 5 volts output and is therefore highly efficient. The output voltage of the regulator is fed directly to the modem and the motherboard. The motherboard then supplies the serial transmitter with 5 volts through the serial cable and the LCD with 5 volts through the unregulated power pin on the microcontroller. The darlington board is fed both the regulated 5 volts and whatever voltage the power supply is providing, as it functions as a form of digital relay (i.e. - it uses one voltage to control the flow of another). To activate the pump, one of the I/O pins on the microcontroller provides 5 volts to the darlington board, which in turn feeds the supply voltage to the pump.

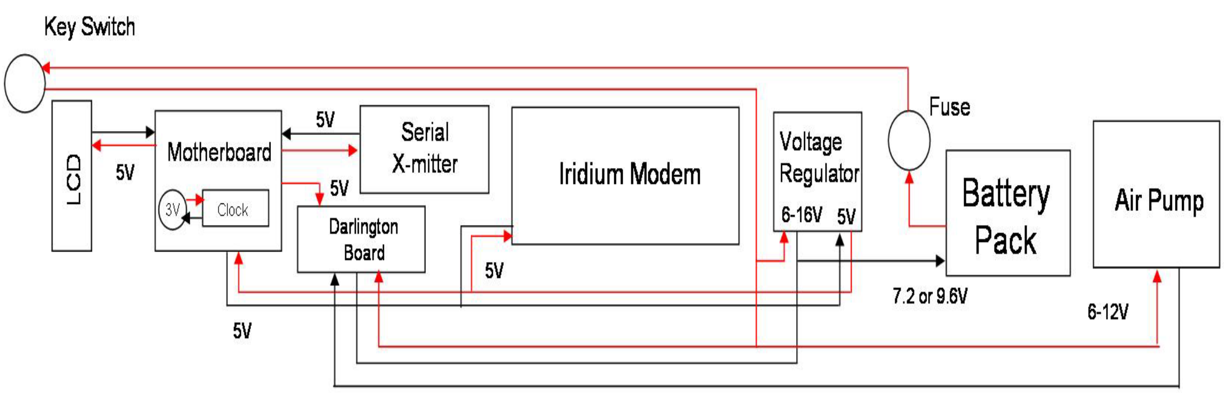


Figure 3.4 - Tumbleweed power conditioning system schematic

3.4. Batteries and Battery Packs

The factors affecting battery choice were numerous and varied. Because the Tumbleweed currently under development is planned for deployment in Greenland, if not Antarctica, battery choice is closely linked with the expected ambient temperature range. In addition, the deployment could last anywhere from 1 to 10 days, with no hope of recovery, necessitating the use of long-life batteries. The weight and number of batteries are given special consideration in order to maintain a balanced load in Tumbleweed such that it rolls along a preferred axis. The maximum weight and number are also affected by the limitation placed on the position of the batteries by the requirement that the modem antenna must remain centrally located to ensure good signal quality. The modem itself experiences current spikes over 4 amps, a level that most batteries are not capable of providing. In addition, it was desirable to find batteries in common sizes, such that readily available battery holders could be utilized.

As previously mentioned, it was also necessary to find both rechargeable and primary (non-rechargeable) systems that could meet all of the above requirements. For the rechargeable system, D - cell nickel metal hydride (Ni-MH) batteries were selected. Two Ni-MH packs were ultimately constructed - one consisting of eight Ni-MH batteries in series (9.6 V, 8Ah) and one consisting of three parallel packs of eight in series (9.6 V, 24Ah) (see Figure 3.5). Both packs were successfully field tested on several occasions. The Ni-MH systems proved to be consistent, long lasting, but extremely heavy.

The primary system proved to be quite a challenge. While lithium batteries are the clear choice when it comes to power density, weight, and operating temperature, there are not many lithium batteries available able to handle the current spikes induced by the modem. After an exhaustive search, one manufacturer was identified that produces a line of lithium batteries whose specs meet all of the requirements. Even though the lithium batteries are the same physical size as the Ni-MH, they operate at more than twice the voltage. To accommodate the lithium batteries, it was therefore necessary to construct dedicated battery packs. Two packs were ultimately built and successfully field tested – one consisting of two parallel sets of two batteries in series (7.2 V, 26 Ah) and another consisting of twelve parallel sets of two batteries in series (7.2 V, 156 Ah) (see Figure 3.5). Of note is the fact that the small lithium pack (4 batteries in total) is superior to the large Ni-MH pack (24 batteries in total) in terms of operating lifetime.

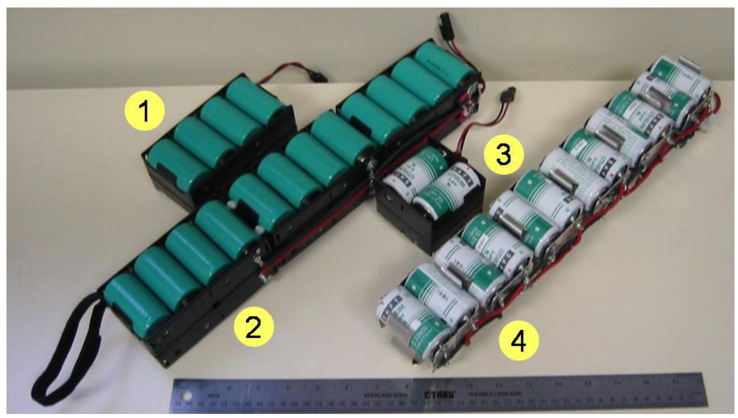


Figure 3.5 – (1) Small Ni-MH Pack (2) Large Ni-MH pack (3) Small Lithium Pack (4) Large Lithium Pack

3.5. Pump System Design and Construction

In the long range deployment scenario planned for Greenland, Tumbleweed will experience a significant change in altitude (the Greenland icecap is dome shaped). If measures are not taken to counteract the loss of internal pressure occurring from altitude change, not to mention small leaks, the rover could deflate to the point that it is no longer possible to roll, well before reaching its final destination. To combat this phenomenon, it was decided to include a small pump in the central tube to actively inflate Tumbleweed during the course of its journey.

The pump is mounted on a the same composite pallet as the electronics (see Figure 3.6). The pump is controlled by the darlington board. The darlington board allows the pump to be powered directly by the battery pack, but switched on and off via the Stamp microcontroller. The pump feeds air out of the central tube and into the membrane via a silicone tube. Threaded into the central tube is an adapter that prevents air from leaking out of the membrane.

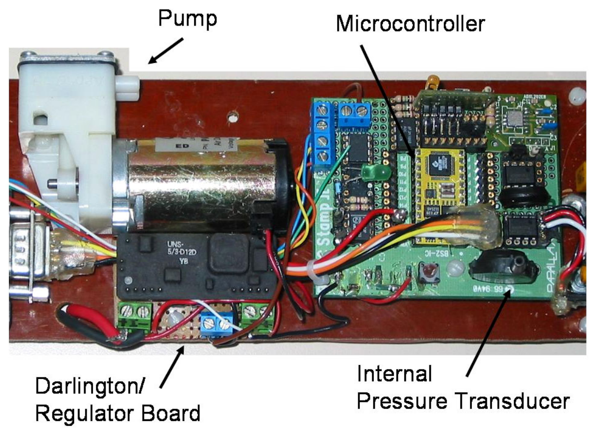


Figure 3.6 – Tumbleweed's pump system.

A second pump/valve system was designed, but never implemented, that would actively inflate and deflate Tumbleweed at a relatively high rate. The Martian Tumbleweed concept calls for the ability to inflate and deflate the membrane as a means of controlling its speed. To test this capability, a pump and valve system that will fit in the central tube of Tumbleweed was designed (see Figure 3.7). This system was to be controlled via the onboard Stamp microcontroller. It was envisioned that eventually the user would be able to command Tumbleweed (via the ground station software) to travel at a certain speed. The microcontroller would then compare this desired speed with the actual speed, as determined via the onboard GPS unit. If the actual speed is not within a certain range of the desired speed, Tumbleweed would then automatically adjust its level of inflation (i.e. – inflate to speed up, deflate to slow down).

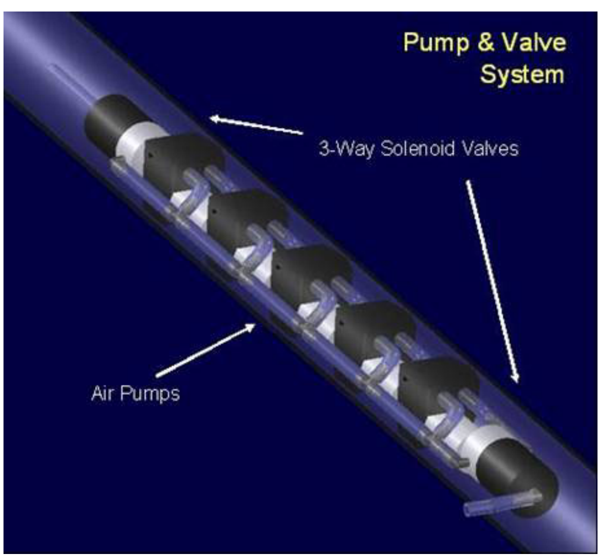


Figure 3.7 – Velocity tracking pump and valve system concept.

An extensive search was undertaken for pumps and valves that are low power, low cost, and fit within the central tube. Several candidate pumps and their distributors were identified. A key parameter noted was the volume of air that the pump can deliver in a given time. This value should be reasonably high to give the feedback control system any viability, as the conditions that Tumbleweed operates in can be extremely dynamic. The most promising pumps are those that are typically used in the battery operated, portable blood pressure monitors. In addition, several solenoid valve manufacturers were identified. The incorporation of two solenoid valves would enable the pumps to be operated in a single direction for both inflation and deflation (see Figure 3.8).

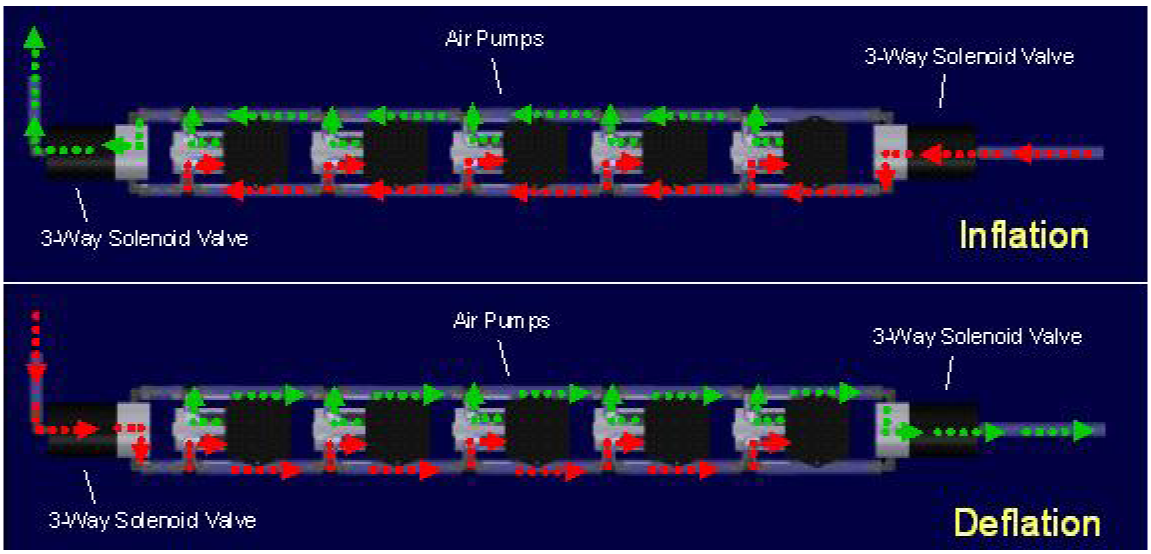


Figure 3.8 – For the velocity tracking system, the pumps are operated in a single direction. The solenoid valves cycle to allow for inflation and deflation through a single port in the central tube of Tumbleweed

4. Ground Station Software Development

4.1. Introduction

To facilitate the field testing of Tumbleweed, as well as the long range Greenland and Antarctic deployments, highly functional, robust, user-friendly ground station software was developed. This software was written in the LabVIEW (Virtual Instrument Engineering Workbench) development environment. The LabVIEW ground station software as written, allows for near real-time data processing and distribution via the internet.

LabVIEW Overview

LabView is based on a graphical programming language called 'G-code.' Unlike a conventional text-based language (FORTRAN, C, etc.), G uses graphical symbols to describe programming actions. The basic files that are created by LabView are called Virtual Instruments (VIs). VIs consist of two main components: the front panel and the wiring diagram (Figure 4.1). The front panel is the user interface and the wiring diagram is the source code. Programs in LabView are constructed much like flow charts, where wires connect one node to another, with each node executing a particular function. Unlike traditional text-based languages which rely on top-down design (i.e. – they execute line by line), LabView execution order is established by data flowing from one node to the next, through the wires and in all directions. LabView is particularly adept at interfacing with external devices. Because all of the equipment that comprises the ground station uses serial communication and the fact that there was a very small amount of time available in which to develop the software, LabView was determined to be the ideal development environment for this application.

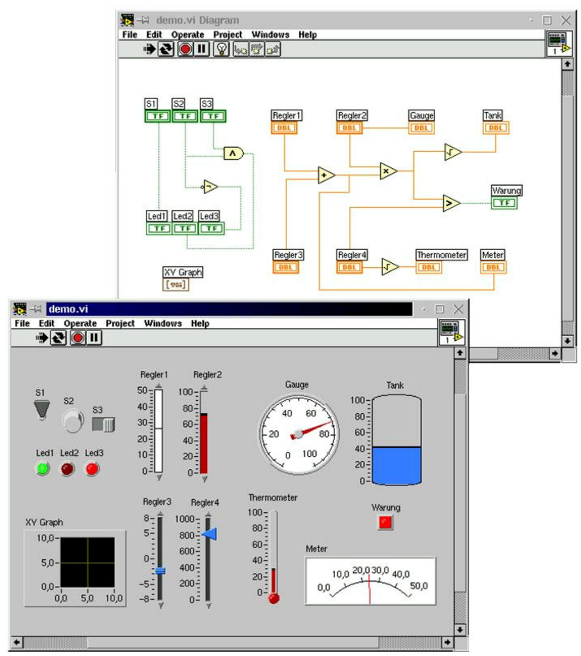


Figure 4.1 – The LabVIEW environment. Top: The wiring diagram. Bottom: The front panel.

4.2. Ground Station Features

The Tumbleweed ground station software accomplishes the following:

→ Awaits a call from Tumbleweed. When a call comes in to the ground station computer, the LabView program answers the phone and sends an acknowledgement back to Tumbleweed. Following acknowledgement, Tumbleweed begins sending the data that it acquired since the last upload.

  This feature is accomplished with the Auto Answer subvi. This vi send commands in the Hayes AT command set (the industry standard for modems) to the ground station modem. The commands sent ensure that the modem automatically answers any incoming calls.

→ The LabView program then receives each data transmission and appends it to a file. The program also does a simple sum check to ensure that all of the data was received. Currently, the sum check is not being used. However, in the future it could be implemented. In which case, in the event of corrupt or missing data, the LabView program would send a message to Tumbleweed and request that the data be sent again.

→ Once the data has been properly received it is posted on the main screen of the LabView program. This screen only displays the most recently acquired data (see Figure 4.2).

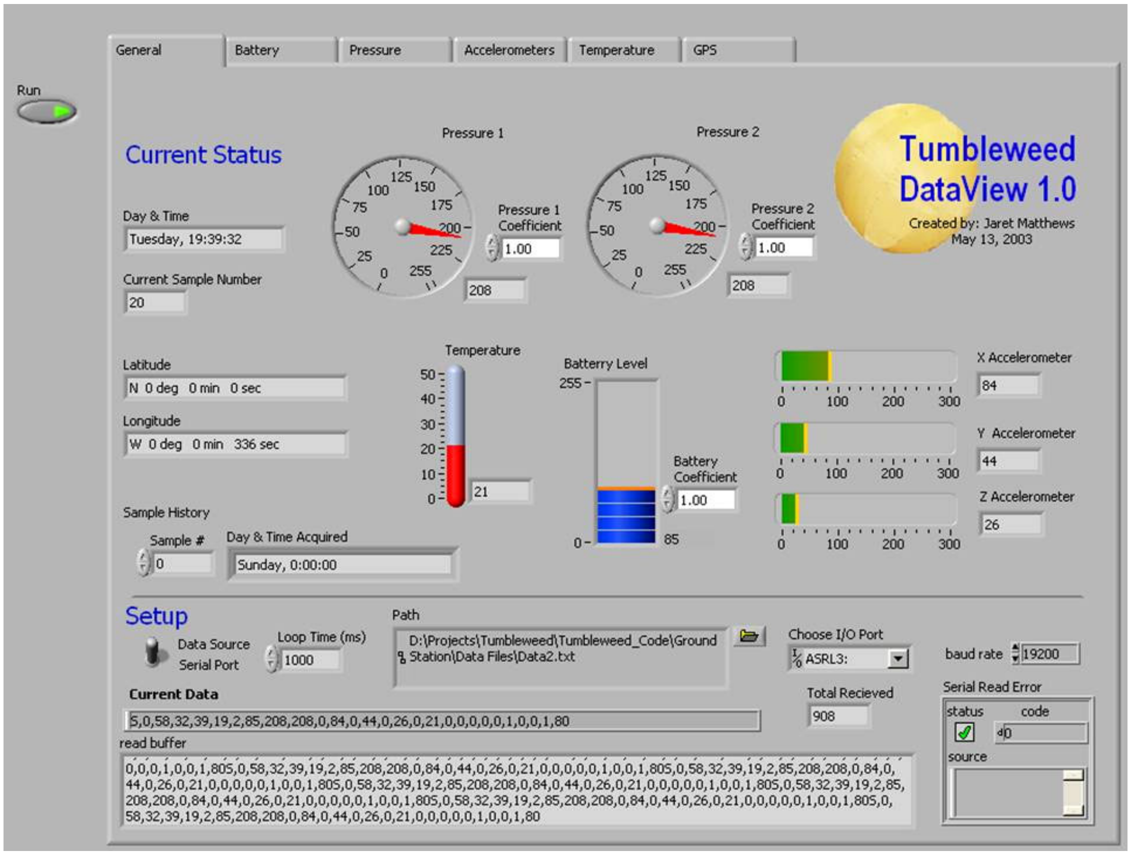


Figure 4.2 – Main screen of the ground station software

The main screen is comprised of two parts. The top two-thirds of the screen is labeled “Current Status” and is reserved for displaying the most recently acquired data, while the bottom third is used for setting up the program and is hence labeled “Setup”.

The “Setup” Field

The setup portion contains several important components. The first of which is the “Data Source” toggle switch. In the switch’s up position, the program looks for incoming data at the chosen I/O port (usually the RS-232 port). In the down position, the program looks to a subvi called “Generate Data” as its data source. The “Generate Data” subvi emulates the data strings normally created by Tumbleweed (i.e. S, 255, 255, 255, 255, etc.). This capability is quite useful for debugging the software.

The second important component of the setup field is the com port settings area on the lower right side of the main screen. This area is responsible for controlling the settings for communicating with the modem. The com port is selected in the

“Choose I/O Port” drop down menu. This menu contains a list of all of the available ports on the computer. This menu is automatically updated with the available ports on the computer that the software is running on. In other words, the user can run the software on another computer and the menu will automatically update to reflect the settings of the new computer. In most instances, the user would select the internal modem port or the RS-232 port that is connected to an external modem. To the right of the “Choose I/O Port” menu is the “Baud Rate” field. Ostensibly, this field is used for setting the baud rate at which the program communicates with the modem. The default for this field is 19200 bps, the required rate for Iridium modems. The field just below the com port settings area is the “Serial Read Error” cluster. This display indicates if there is a problem communicating with the selected port and gives a brief description of the likely problem.

The central area of the setup portion of the main screen contains several important fields. First is the “Path” field, which enables the user to select the location and name of the text file to which he or she would like the data strings written. Before running the program, the user clicks on the file icon and a dialog box appears in which the desired path can be selected. If the user does not select a path before initiating the program, they will be prompted to do so and the dialog box will appear automatically. If the user selects an already existing file they will receive an error prompt asking if they would really like to overwrite the existing file. Each time a complete string of data arrives at the serial port (see following discussion on ensuring a complete string) it is appended to the text file selected in the path menu.

Just below the “Path” field is the “Serial Read Buffer”. This field displays the unformatted data at the selected serial port as it arrives. The serial read buffer only displays the data at the serial port that has accumulated since the last execution of the main loop of the program (i.e. – since the last time the serial port was read from). Because the data in the buffer can arrive in randomly truncated chunks (i.e. – 255, 255, S, 255, 255……..rather than the expected S, 255, 255, 255…), the program must split and correctly reassemble the chunks into a form that the rest of the program can understand. The result of the operation is displayed in the “Current Data” field just below the serial read buffer field. The “Current Data” contains the unformatted string of data that is being displayed in the top portion of the main screen.

The sum of the numbers contained in the current string of data is displayed in the “Total Received” field. This number is not currently used in the operation of the program. The total received value however, could be used in the future to ensure that the data received matches the data sent from Tumbleweed. In this scenario, the total received value could be compared with an expected sum sent by Tumbleweed. If the two numbers did not match, the LabView program would send a command back to Tumbleweed requesting that the data packet be resent.

The “Loop Time” numeric control is the final important component of the setup field. This value represents the time in milliseconds between executions of the main loop of the program. It therefore represents how often the serial port is read and new data displayed. This value allows for the user to set the speed at which the program runs and therefore indirectly the amount of processing power required of the ground station computer. The default value for the loop time is 500 ms. The value should not be set much lower than this to ensure sufficient time for communication with the modem.

The “Current Status” Field

As mentioned above, the “Current Status” field is to used to display only the most recent string of data. To display this data in a manner which is coherent to the user, the unformatted data must first be parsed. To do so, the program passes the unformatted data to a subvi called “Split String”. The split string vi breaks up the unformatted data string into its respective parts and sends them back into the main part of the program. The unformatted string starts with an “S” and is followed by a series of comma delimited bytes. Parsing the string correctly is therefore fairly straightforward and is simply a matter of formatting the string to an array (by searching the string and splitting it after each comma) and indexing that array at each byte.

After leaving the split string vi the data is in a form that can easily be manipulated and displayed throughout the rest of the software. The remainder of the program uses what are called local variables to check the status and thus work with each byte of data leaving the split string subvi. Local variables simply track the value of their parent variable throughout the program without having to be directly wired to it. Using local variables enables a significant reduction in the amount of wires used on the wiring diagram.

To arrive at the formatted data displayed in the current status field, many of the bytes must fist be manipulated. The first of which are the bytes representing the time and date of the acquired sample. Tumbleweed’s real-time clock sends this information in the following manner:

seconds (0-59), minutes (0-59), hours (0-23), days (0-5)

To format this data such that it is displayed “Day, hh:mm:ss”, it is first necessary to check the seconds and minutes bytes to see if they are less than 10, in which case a zero must be inserted prior to the value to ensure correct formatting (i.e. rather than Sunday, 5:5:5, it will display as Sunday, 5:05:05). The hours byte is used as is and the days byte is converted to a text representation, where 0=Sunday, 1=Monday, etc.

Tumbleweed’s thermometer is digital, therefore its value can be directly represented without significant manipulation. It is however, sent as two bytes (high and low) that must be recombined before being displayed properly as one value. Using the join numbers function in LabView, the two 8-bit numbers representing temperature are combined into a single 16-bit number. This is not the case however for the two pressure transducers or the battery level. Each of these values are arrived at by way of an analog to digital converter on Tumbleweed's motherboard. It is therefore necessary to calibrate the digital values to their actual values. This is done by simply multiplying the digital values by a coefficient. The coefficient for each value is set in the numeric control located next to each of the graphical displays. The displays present the calibrated value to the user in an easy to read, quickly viewable format, such as the type of gauges that most users are familiar with. The calibrated numeric value for each gauge is simultaneously displayed to the right of the respective display.

The most recent value of each of Tumbleweed's accelerometers is displayed in a similar manner. Here again, because the instruments are digital, the resulting data does not require significant manipulation. They do require recombination of the high and low bytes, just as is done with the temperature values. While not yet completed, some work was done to display the accelerometer as one point on a three dimensional plot (i.e. the X, Y, and Z values are presented simultaneously with one point). Though some progress was made in this regard, the final display is simply a set of three fill slider bars and their respective numeric values, each representing one axis.

The incoming GPS data requires little manipulation to arrive at its final displayed form as it is seen on the "Current Status" screen. However, to ensure that it plots correctly (described later) requires more treatment. The data arrives in the following manner:

> N/S latitude (0 or 1), degrees latitude (0-90), minutes latitude (0-59), seconds latitude high byte (0-255), seconds latitude low byte (0-255), E/W longitude (0 or 1), degrees longitude (0-180), minutes longitude (0-59), seconds longitude high byte (0-255), seconds longitude low byte (0-255)

The formatted data is displayed in the following manner:

> N, ##deg, ##min, ##sec
> E, ###deg, ##min, ##sec

To begin with, the program must turn the numeric representation of the hemisphere into a text representation. This is accomplished fairly easily with a case structure in which 0=North; 1=South and 0=East; 1=West, similar to the way in which a byte representation is changed into a text value of the day of the week. The high and low bytes for the second values must be recombined to arrive at a number between 0 and 65535 (a 16-bit number). This is done in the same manner as the temperature and accelerometer values described above.

Archived Status Pages

→ The LabView program also plots all of the received data on various graphs (see Figure 4.3). The graphs are accessed by pressing the tabs pertaining to the various types of data on the top of the interface. For battery level, temperature, accelerometer, and pressure, the data is simply plotted versus sample number. At the bottom of each screen, the user may click a digital control box to see the time and date pertaining to each sample number.

The plots are updated as each new point comes in, but only if the user is viewing the plot at the time. This prevents wasting processor resources on plotting information that is not of interest to the user. The sample number axis (the x axis) is auto-scaled such that the data is presented in the most readable way possible regardless of the number of samples that have been acquired. The sample value axis (the y axis) is set to a range of expected numbers. The max and min values of this axis can be changed without stopping the program in the event a value is received that is out of the expected range.

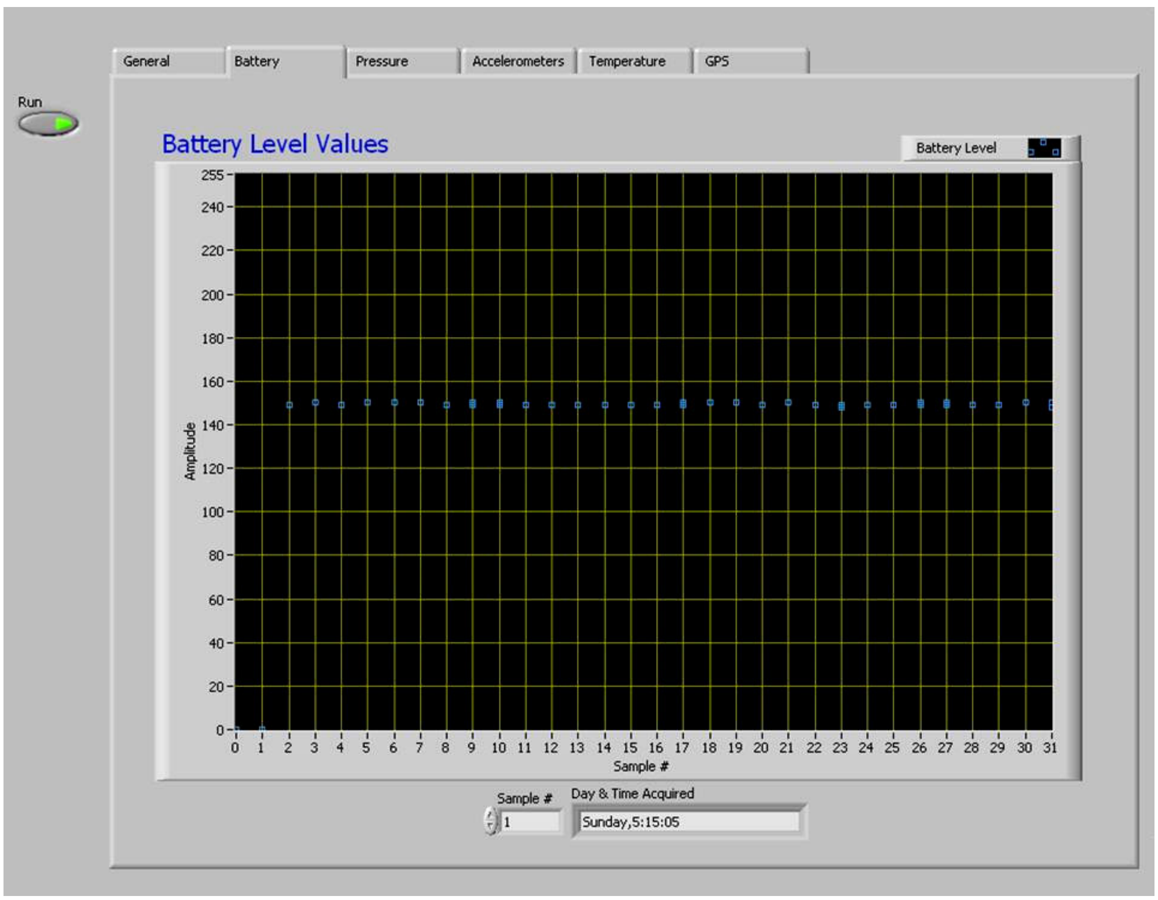


Figure 4.3 – An example of the various data plots available with the ground station software

Again, work was done to plot all of the accelerometer values as single points on at 3-D graph. However, due to insufficient time, the accelerometers are currently plotted as separate lines on the same graph. The lines are differentiated by color.

→ The GPS location is plotted over any georeferenced satellite image, DEM, or map available for the area in question. This is accomplished by simply dropping in an image (via copy and paste) to which the latitude and longitude of the corners of the image are known. This image is placed behind the GPS plot using the order command on the LabView toolbar. The corners are aligned with the arrow keys and the mouse, and the max and minimum values of the graph axes are adjusted to match the corner values of the image. The position and ground track of Tumbleweed can thereby be monitored with respect to a region's topography, groundcover, or any other relevant information. Again, the user may scroll through the acquired data to determine Tumbleweed's location at any point prior, by clicking on the digital scroll at the bottom.

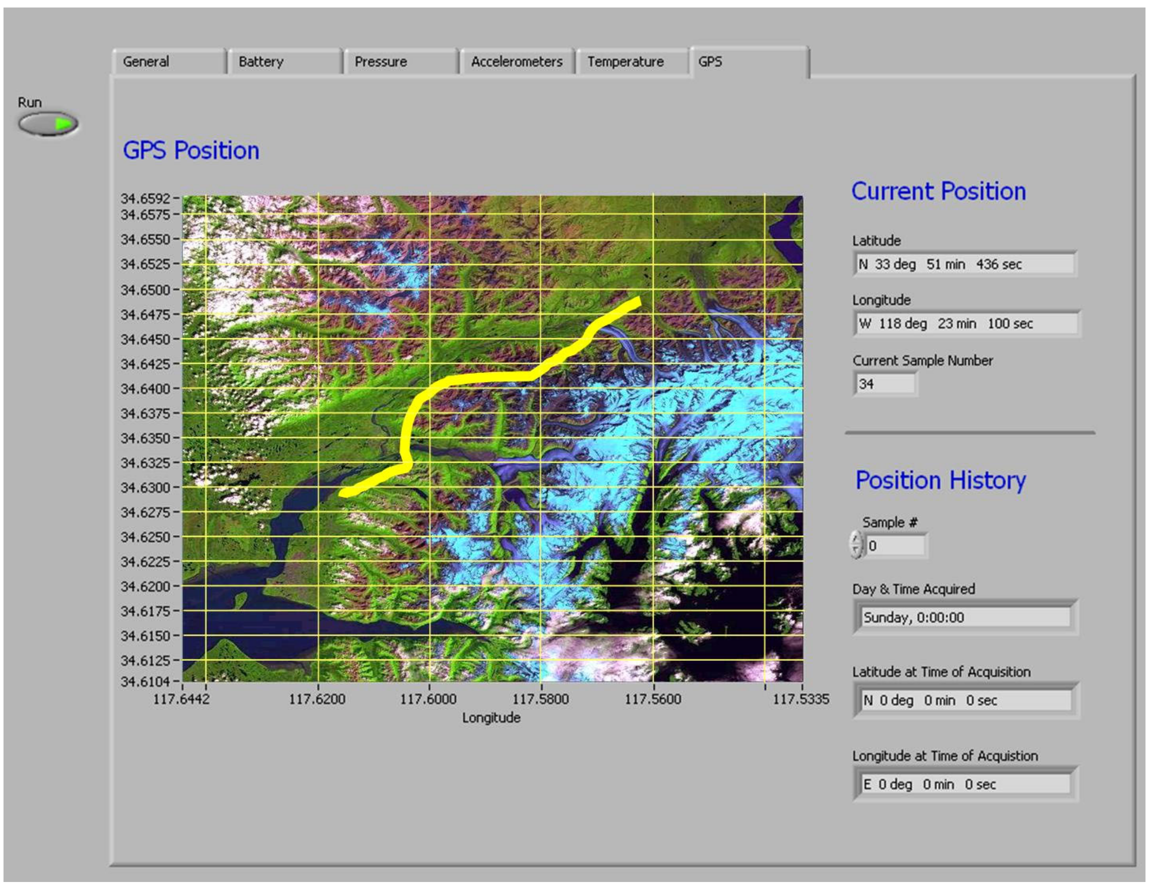


Figure 4.4 – The location of Tumbleweed is plotted over any georeferenced image.

As mentioned, the GPS data undergoes significant alteration before being plotted. The position information for the rover arrives in the form N/S, deg, min, sec (see previous discussion) and must be changed into a single number representing the same in order to be properly displayed. To do so, the seconds value is divided by 3600 and added to the hour value. Similarly, the minutes value is divided by 60 and added to the hour value. For the latitude, if the location is in the southern hemisphere, the total position value is changed to a negative number. Resulting in a range of latitude numbers from -90.00000 to +90.00000. Similarly, if the longitude position is in the western hemisphere, the total position is changed to a negative number, resulting in longitude values from -180.00000 to +180.00000.

→ When run on a computer with internet access, the LabView program allows other interested parties to view and even control the ground station from anywhere in the world. Remote users simply type in the URL of the host computer and are immediately able to see the LabView program live. This capability could allow, for example, people at JPL and NASA headquarters to simultaneously watch the Tumbleweed data live. In addition, the ability for anyone to have immediate access to the data is favorable to both the scientific community as well as the public at large. It is not hard to image a significant amount of public interest being generated as live images stream in from an Antarctic-crossing Tumbleweed.

This feature is surprisingly easy to implement in the LabView environment. The latest versions of LabView (6i and 6.1) come with a built in web server. The operator need only to start this server and export the main vi as an html file. This function does not even require that remote users have LabView installed on their own computers. In this case, after logging on to the ground station’s website, their computer will automatically begin downloading and installing the required applet. The appendix of this document contains a more detailed discussion on operating the LabView web server.

4.3. Software Testing

While being developed, the ground station software underwent several rounds of testing. This testing was incremental and driven largely by the readiness of the hardware. The software test program was designed to test the viability and functionality of the software while giving minimal opportunity for failure induced by outside sources.

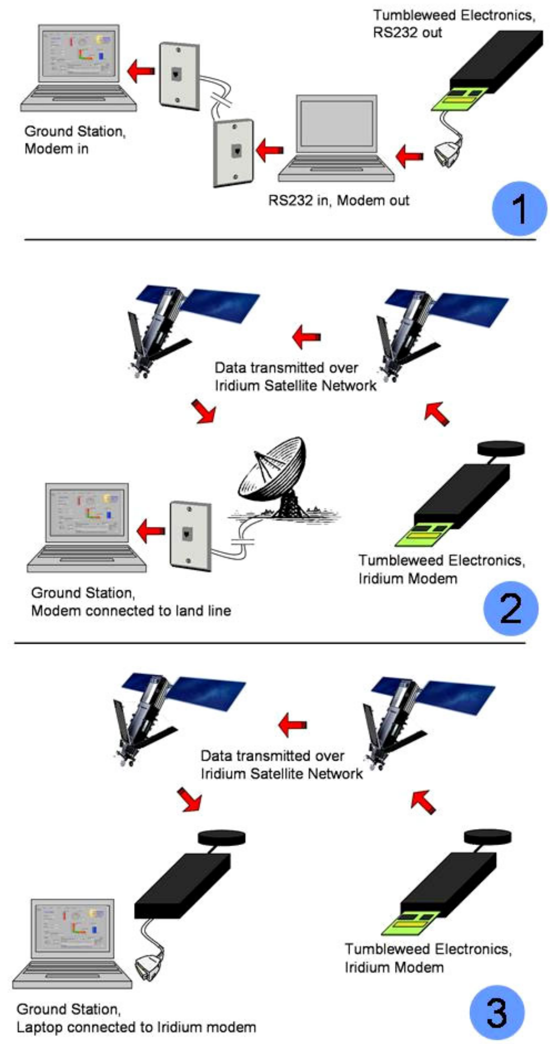


Figure 4.5 – The three major phases of software testing

As Figure 4.5 depicts, the software was tested in three major phases. The first phase (part 1 of Figure 4.5) of testing involved the use of a second laptop as an interim modem. In this setup, Tumbleweed's onboard Iridium modem was used to acquire GPS data but not to make phone calls. The data was passed from the Tumbleweed motherboard to the second laptop via RS232 port. There was a simple LabView program running on this computer that dialed a number and passed the data from the RS232 port directly to the internal modem. This modem dialed the ground station computer via land line. Here again, the internal modem on the ground station computer was used to receive the data. This test setup eliminated failures caused by using the Iridium network.

After successfully operating the ground station software in the first mode, a second, more complicated mode was attempted (part 2 of Figure 4.5). This round of testing involved the use of Tumbleweed's onboard Iridium modem to acquire GPS data and make phone calls. This phase of testing went quite smoothly and soon a third and final mode was attempted (part 3 of Figure 4.5). The final round involved both an Iridium modem on the Tumbleweed and ground station sides for making and receiving calls. While the performance of the system was quite erratic at first, adjustments were made and eventually confidence in the software was gained. Most of the inconsistency can be attributed to the occasional loss of signal with the Iridium satellite network.

5. Ground Station Hardware Development

To support the field testing of Tumbleweed, a portable ground station was constructed. This ground station contains a laptop, Iridium modem with integrated GPS, an omni-directional Iridium antenna, a GPS antenna, a serial transmitter, a folding antenna mast, a 300 W 12VDC to 120AC converter, a 120 VAC to 5VDC converter, a pulse modulated 5V voltage regulator, and protective foam. All of the above components pack nicely into a weatherproof, rolling hard case. The ground station can operate off of either 12VDC power (for remote applications) or 120VAC (when wall power is available). The folding antenna mast and its mounting bracket were constructed from tube and stock aluminum in the JPL machine shop. This ground station has performed successfully on several occasions in the field. Because it uses an Iridium modem to receive data, the ground station can operated anywhere in the world.

Figure 5.1 – Ground station hardware. (clockwise from left) 1. The fully deployed state. 2. The equipment stowed. 3. Ready for transport to the field.

6. Field Test Program

Over the last three months, Tumbleweed has undergone a rigorous field test program. This program consisted of a total of four deployments, including two to the harsh Mojave Desert (see Figure 6.1).

Figure 6.1 – Testing Tumbleweed in the harsh conditions of the Mojave desert

The first deployment was to the Lucerne dry lake bed. This principal goal of this test was to verify the roll characteristics of Tumbleweed after the replacement of the central tube. It was feared that the larger and consequentially heavier tube and electronics package would adversely affect the way in which Tumbleweed rolls. Prior to the integration of the new tube, Tumbleweed had a tendency to auto-correct its orientation such that it would roll with the tube parallel to the ground. The Lucerne test did raise concerns that Tumbleweed had lost this auto-correcting ability, but ultimately proved the viability of the new hardware.

Lucerne was also the site of the first field test of the ground station software. While the software performed successfully when the electronics package was outside of the nylon ball, full-up rolling tests were made impossible by dying batteries. The most significant result of the Lucerne test was the demonstrated need for portable, easily deployable ground station equipment of the nature described in the previous section.

The electronics package for the Lucerne test was broken into two pieces, with the electronics and antennas on one side and the battery packs on the other. Following this test, it was decided to put all of the components onto a single pallet. This decision was later reversed after subsequent field testing proved the original setup to be more practical.

Two weeks after the first, a second field test was conducted at an abandoned airfield just West of Edwards Air Force Base (still in the Mojave desert). This test was the first with the new ground station hardware, which performed its job perfectly. Several successful data transmissions were completed using new Ni-MH battery packs. A lithium pack was tried as well but proved to be insufficient for making phone calls. Much of the time on this test was spent chasing a problem with the GPS data. This problem would plague subsequent tests as well. As mentioned, a new single electronics pallet design was evaluated. This approach was dropped as a result of antenna and battery placement issues that appeared in this round of field testing.

Two weeks after the second, a third field test was conducted at a baseball diamond in Los Angeles (see Figure 6.2). This location was chosen due to its wide open space and the availability of electricity. This test was the most successful yet and marked the first successful rolling transmission of data using the new hardware, ground station equipment, and software. This test was conducted with the separate pallet and battery setup. Again, using Ni-MH batteries the system performed quite well, but a reworked lithium pack was still deemed insufficient. The problem with the GPS data was also persistent through this round of testing.

Figure 6.2 – Testing the Tumbleweed rover and ground station

The fourth and final field test was conducted one week later. Just prior to this test, it was discovered that the GPS data was lost when the antenna was brought within too close of a proximity to the Iridium antenna. It was therefore decided to move the GPS antenna to the end of the tube, as far away as possible from the centrally located Iridium antenna. This setup worked sporadically and tests on this day were the first in which all of the components functioned at the same time. The ground station software consistently and correctly received incoming calls, parsed, saved, and displayed the data. This test proved the functionality of all of the components as required by the upcoming Greenland and Antarctic tests.

7. Conclusion

The development of the Tumbleweed rover has proven to be an extremely challenging and rewarding experience. A plethora of new components have been designed and fabricated or otherwise procured that have advanced the Tumbleweed system significantly in the last few months. Ground station hardware and software have been developed that will ensure the reliable data acquisition in even the most harsh and remote environments.

While there is much to be done to ensure the reliability of the system under harsh operating conditions, the test program to date has proven to be invaluable. Frequent testing forced quick but informed design iterations that progressed the hardware and software an extraordinary amount in a short period of time. Much has been learned on all fronts, but especially with respect to power systems, component layout, and software architecture. The knowledge gained in the recent development and test programs will certainly contribute to Tumbleweed's success in future, long-range deployments.

Note: The following chapter suggests a possible field deployment location to compliment the other planned deployments. The proposal for this alternate test location was requested as a result of the author's personal experience with the location in question. The test location described represents a unique and extraordinary opportunity to test Tumbleweed rover in an environment resembling both Mars and Antarctica in many respects.

8. Endurance Field Test Location Proposal

The Mars analog environment of Devon Island should be considered for the upcoming field tests of the Tumbleweed rover. The following chapter outlines several benefits that Devon Island and the NASA Haughton-Mars Project (HMP) have to offer to any test program, as well as specific advantages that would be available to the Tumbleweed test program.

Devon Island and the Haughton Impact Crater

Located at 75° N latitude in the Nunavut Territory of the high Canadian arctic is Devon Island (see Figure 8.1). Devon Island has a surface area of approximately 66,000 km^2 (roughly the size of West Virginia), making it the largest uninhabited island on Earth. The easternmost third of the island is covered by an ice cap and the remainder is a barren, rocky, strikingly Mars-like terrain.

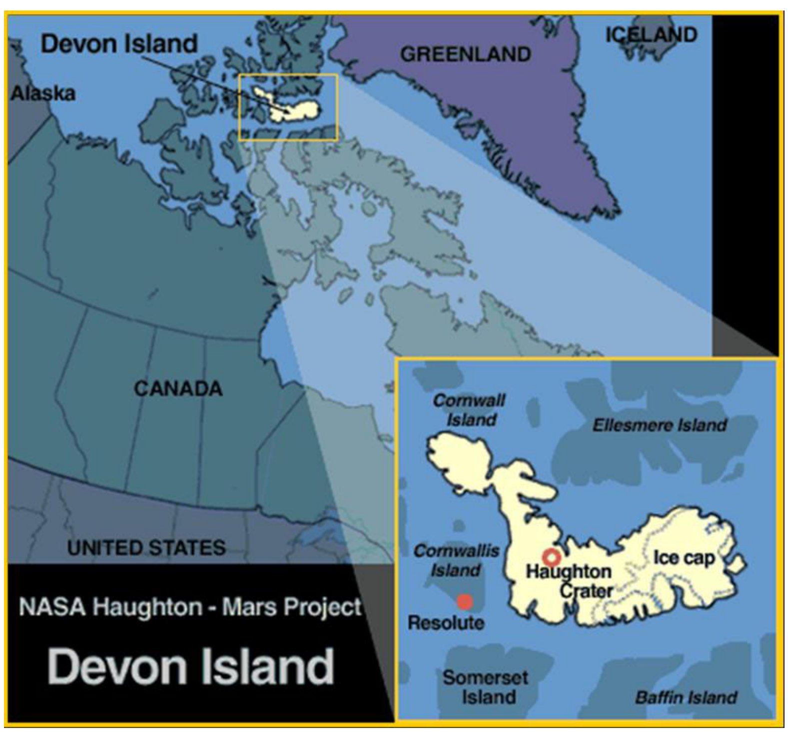


Figure 8.1 – Devon Island and the Haughton Impact Crater

Just left of center on Devon Island is the 20 km diameter Haughton impact structure. This 23-million-year-old crater is one of the best preserved in the world and is the only one located in a polar desert environment. This already sets Devon Island apart as a Mars analog, as Mars is essentially a polar desert that is littered with craters. In addition, the temperature is similar to those experienced on Mars at peak times, at mid-latitudes, in the middle of summer (i.e. – its Mars on a very, very good day). The terrain is ground-ice-rich, impact-generated rubble and experiences a high UV-flux, as is the case on Mars (see Figure 8.2). There have been other morphological similarities identified as well, from snow-melt formed gullies to polygonal patterned ground to impact-generated hot springs.

Figure 8.2 - Almost identical terrain is found on Devon Island (left) as is found on Mars (right).

The NASA Haughton-Mars Project

In 1996, Devon Island and the Haughton Crater, in particular, were recognized by Dr. Pascal Lee, of NASA Ames and the SETI Institute, as promising Mars analogs (for the reasons mentioned above). Dr. Lee initiated the NASA Haughton-Mars Project (www.marsonearth.org) to study and exploit this region for its inherent scientific value as well as for its remarkable resemblance to Mars.

The HMP camp is situated on the northwest rim of the Haughton crater (see Figure 8.3) and accommodates an average of 15-20 people at any given time (at peaks times it has held more than 50 participants). The field season typically lasts from mid-June to early-August and 2003 will mark the camp's seventh year of operation. After arriving in Resolute on a commercial jet (flown out of Edmonton or Ottawa), participants reach Devon Island via a short skip in a Twin Otter. Much of the cargo is delivered to Resolute from NASA Ames on a C-130 in late May and flown over in several Twin Otter flights at the start of the field season. If cargo is not ready in time for the C-130 flight, the HMP has an agreement with the Polar Continental Shelf Project to receive and store incoming shipments at any time.

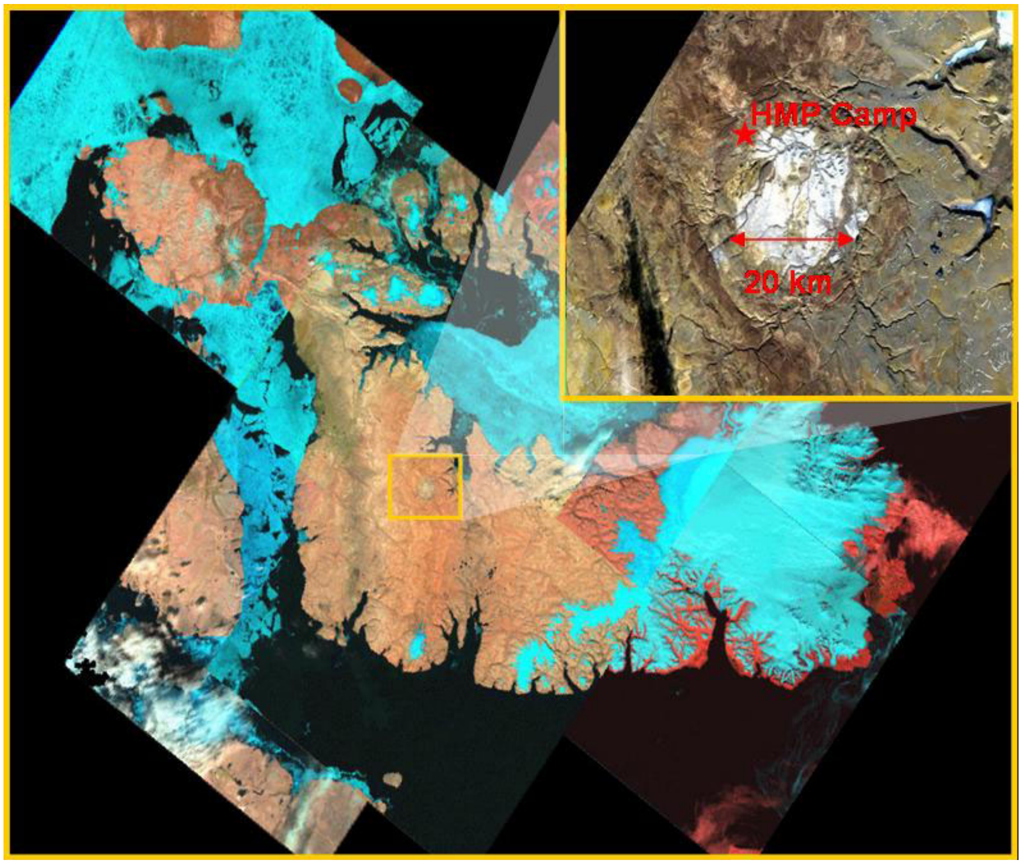


Figure 8.3 - LandSat mosaic of Devon Island and the Haughton Impact Crater.

The HMP has the following facilities/infrastructure available at camp (see Figure 8.4):

- All-terrain vehicles
- Hummer
- Work tents/Lab space (now heated!)
- Generators (some operating 24 hours/day)
- Tools
- High-speed Internet Access
- Weather Stations
- GIS Data
- Showers
- Kitchen/Dining Tent

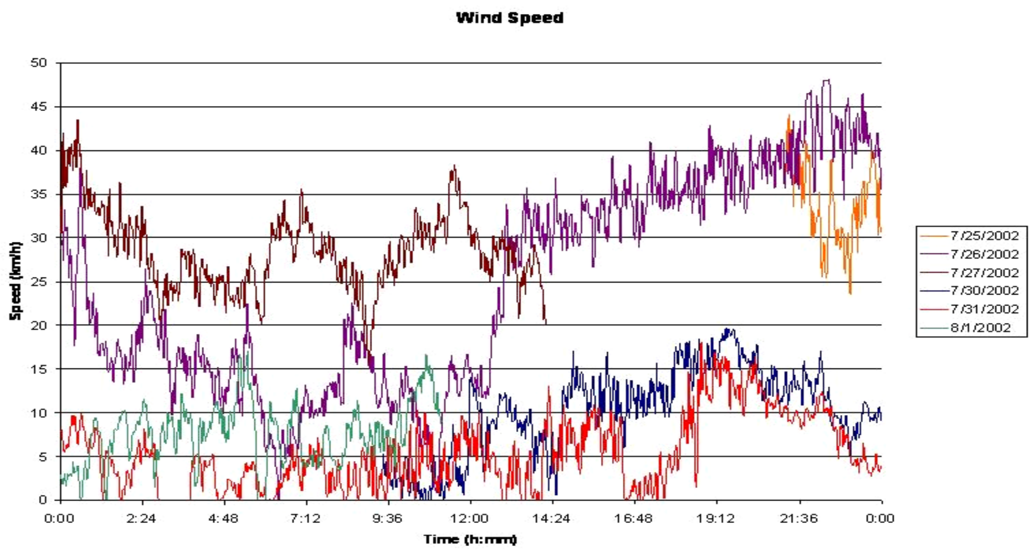
Figure 8.4 – NASA HMP Camp

In addition, HMP has a history of robotics research (see Figure 8.5). The author personally spent the last two summers on Devon Island working with the Purdue University Rover for Mars Analog Research (P.U.R.M.A.Cat) and the Purdue University Web-Operated Rover for Mars (W.O.R.M) and is therefore very familiar with the parties involved, the camp, the island, and the logistics surrounding field testing on Devon.

Figure 8.5 - Robotic field tests conducted on Devon Island (*L to R from Top*) 1. Carnegie Mellon University Helicopter, 2. Dale Stokes Buoyant Observing Balloon (B.O.B), 3. & 4. DARPA Robots, 5. Carnegie Mellon University Hyperion, 6. Purdue University/CSA Web Operated Rover for Mars (W.O.R.M.), 7. NASA Ames MEX-HORSE, 8. Planetary Society Mars Airplane

Devon Island as a test site for Tumbleweed

Tests on Devon Island could serve to evaluate Tumbleweed in both a Mars analog environment (on the Western 2/3 of the island) and an Antarctic analog environment (on the Eastern 1/3 of the island). The extremely large size of the island would enable a long distance test (at least the 200km minimum established for the Antarctic). The available means of transportation (numerous ATV's, a Hummer, and potentially a helicopter) would enable the recovery of Tumbleweed, if one so desired. There is almost always at least a 10 kph wind, with sustained winds up to 40 kph and much higher gusts (see Figure 8.6).



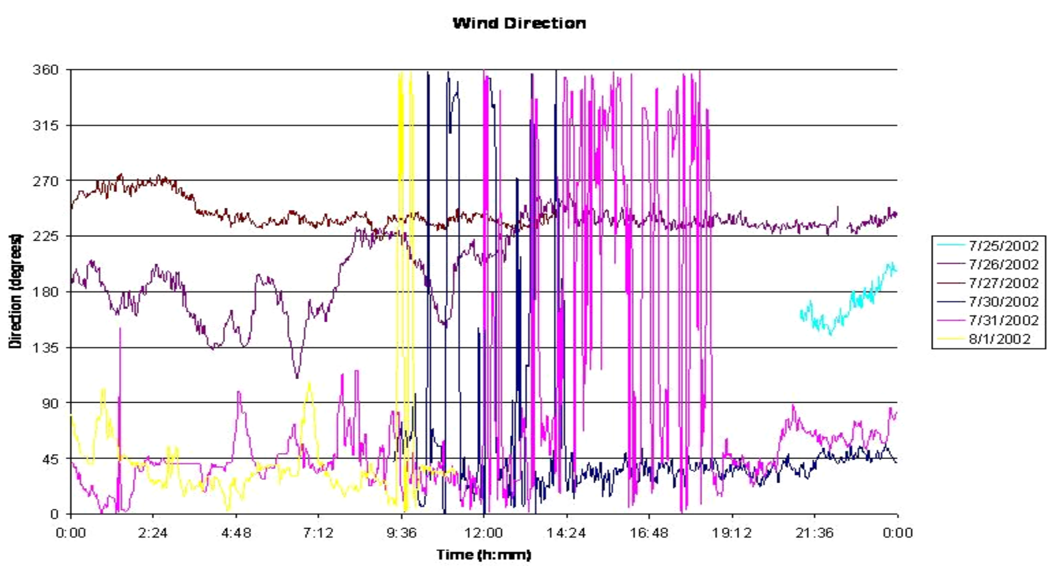


Figure 8.6 - Wind speed and direction at camp over several days in the 2002 season.

References

Jones, Jack A. "Inflatable Robotics for Planetary Applications." 6th International Symposium on Artificial Intelligence, Robotics, and Automation in Space, I-SAIRAS, Montreal, Canada, June 19-21, 2001.

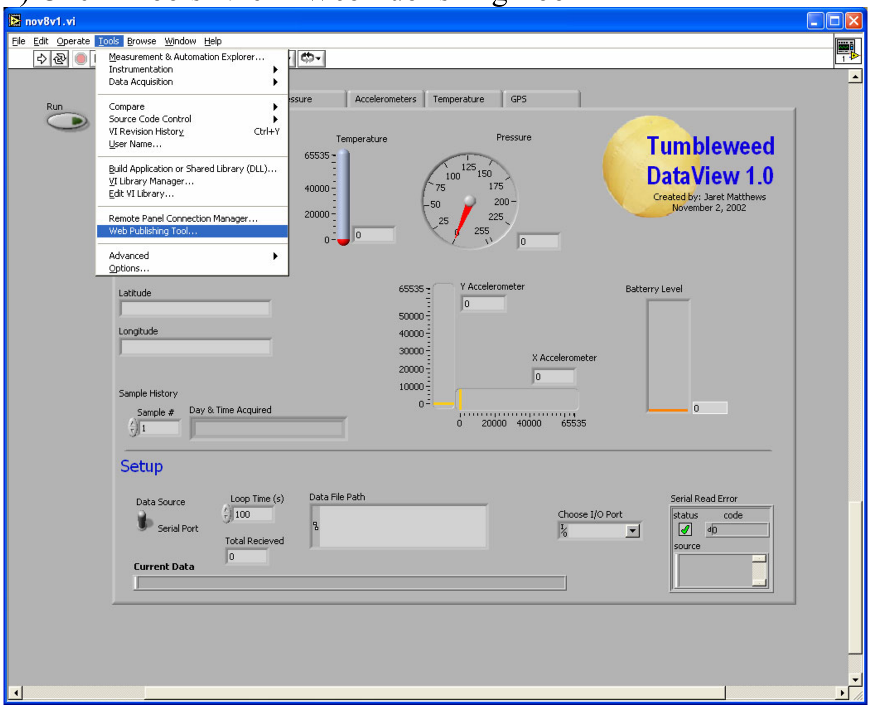
Jet Propulsion Laboratory Press Release. "Spotlight: Exploring Mars: Blowing in the Wind?" Pasadena, California. August 10, 2001.

Appendix

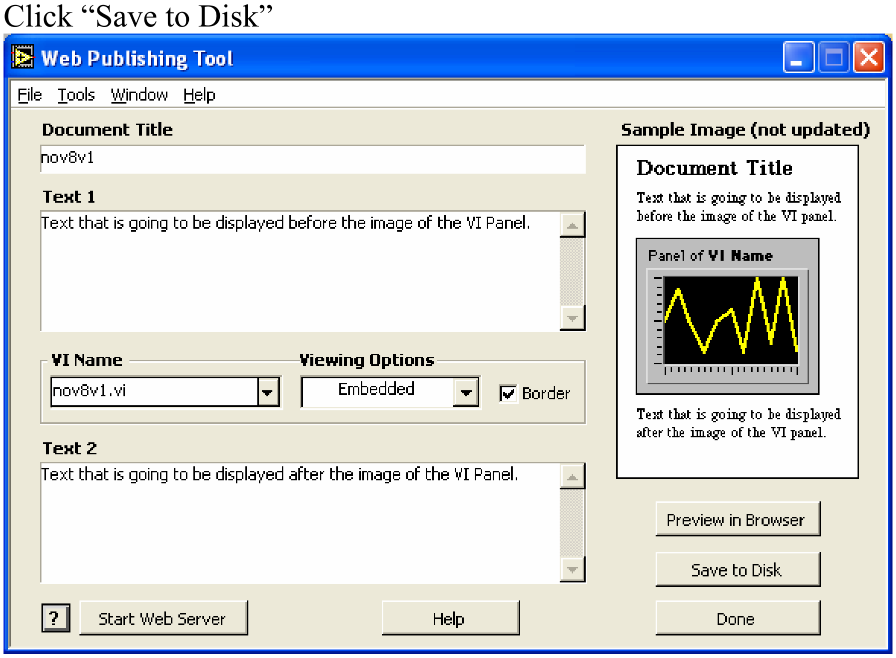
Remote Operation Using LabView 6.1

In LabView 6.1:

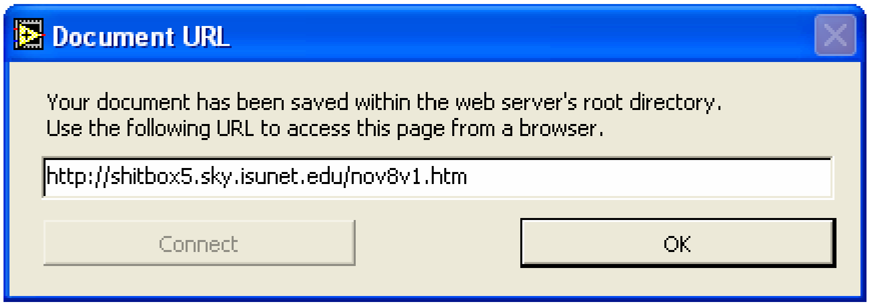
1) Click “Tools” then “Web Publishing Tool”



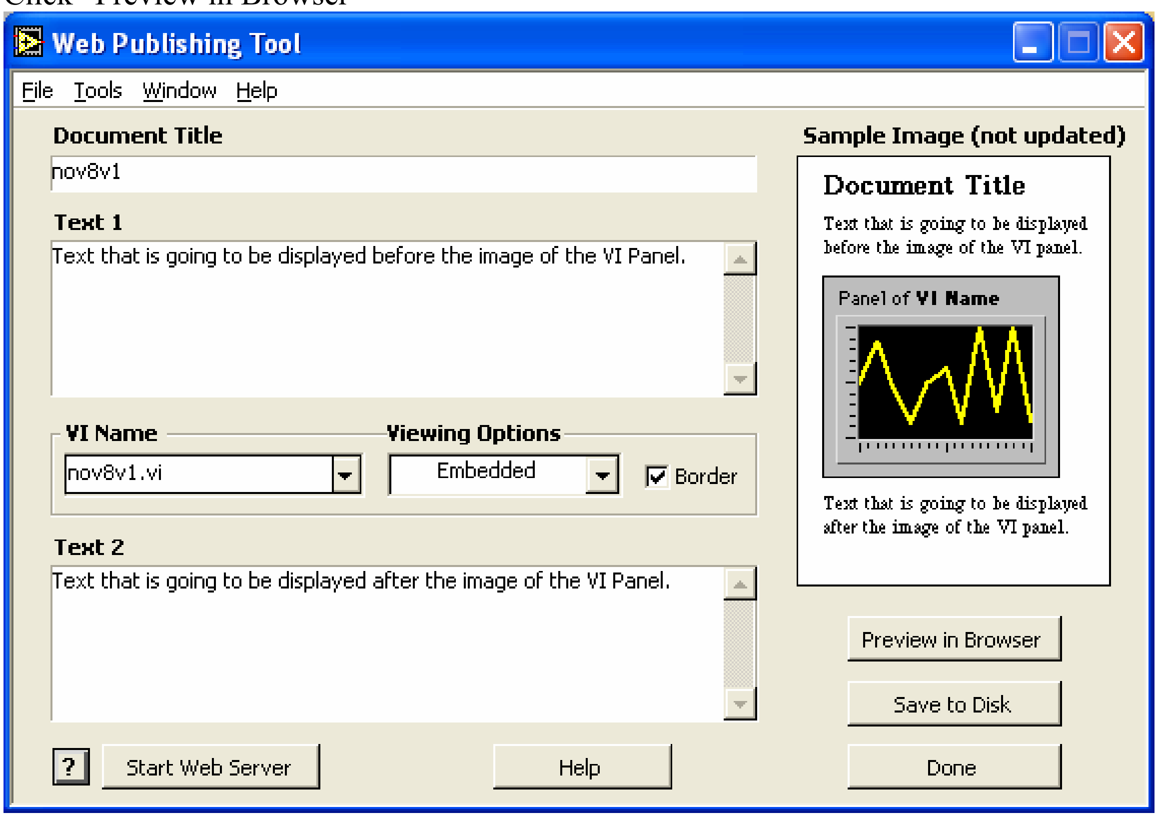
2) The following window appears:
Click “Save to Disk”



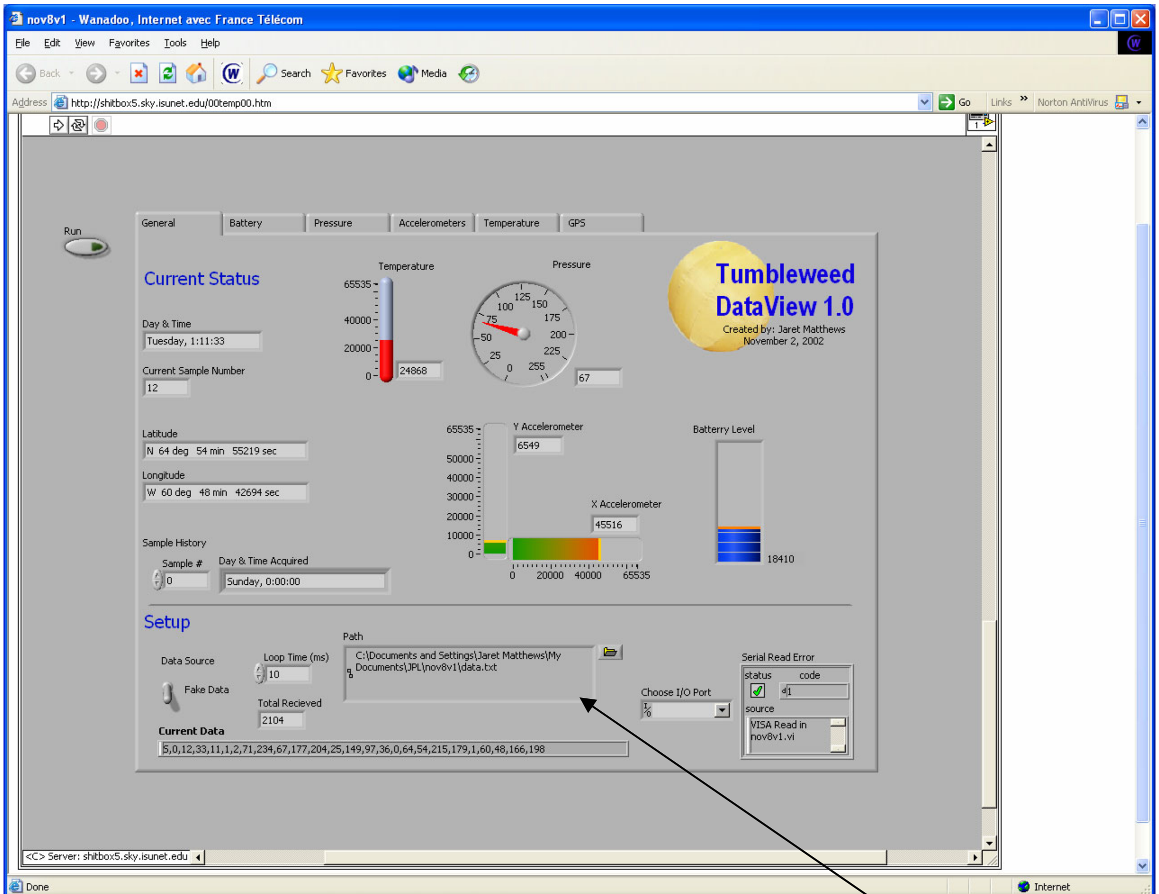
3) The following window appears:
Write down the URL and distribute to anyone that you would like to have access.
Click "OK"

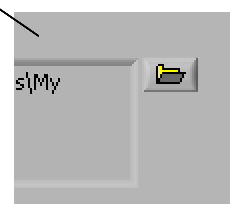


4) Now you are back at this window:
Click "Start Web Server"
Click "Preview in Browser"



5) The following window appears:
Right Click for options
Click on “Request Control of VI”

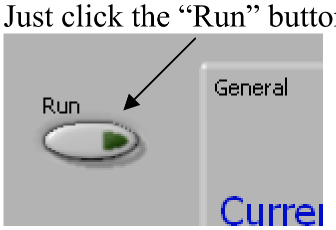


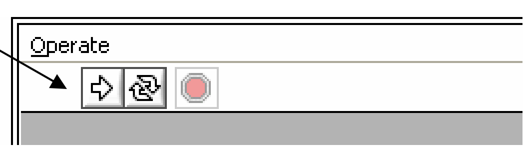


You should receive a message “Control Granted”
!!!!Before you run the vi remotely do the following:
Click on the small folder icon and choose the path for where you want to save the data file. You also need to choose the name of the data file (i.e. – data.txt)

Note: If the vi was already started on the computer running the web server, you don’t need to worry about this b/c you would have already been prompted to choose a file path.

Now you are ready to run the vi remotely.
Just click the “Run” button and the “Operate” arrow





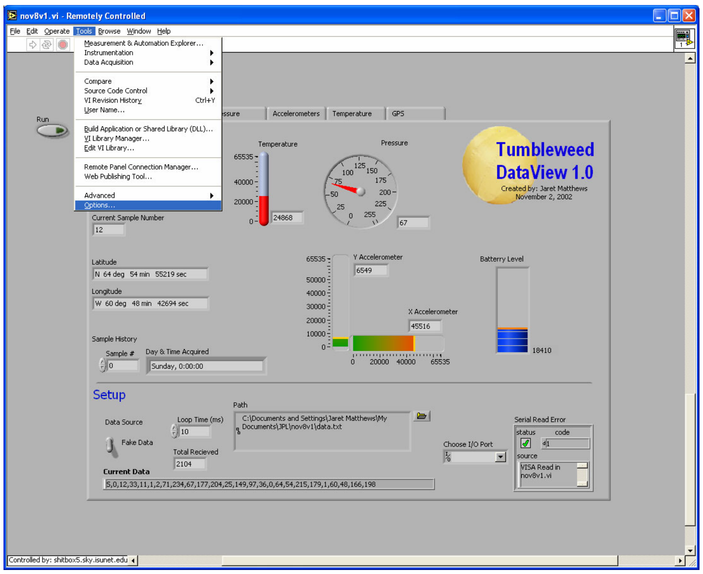
If some tries to connect to the page and they don't have LabView installed on their computer, it will automatically get the required applet from the following site:

ftp://ftp.ni.com/support/labview/runtime/windows/6.1/LVRunTimeEng.exe

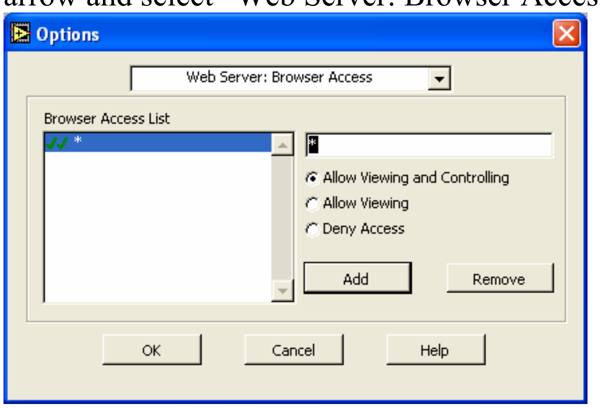
It is quite a big file and can take some time to download. So have them be patient!

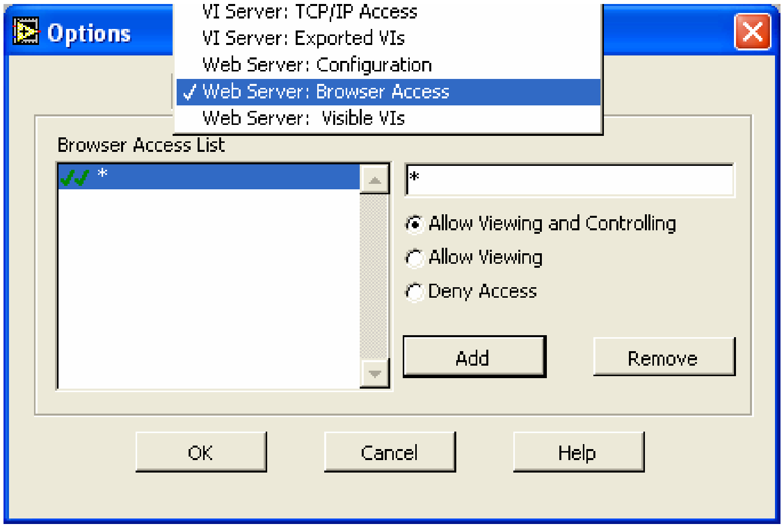
Setting up who can see or control the Vi…

1) Click "Tools" then "Options…"



You will see the following window (if it doesn't look like this then click on the menu arrow and select "Web Server: Browser Access"





As you can see this is also where you configure the Web Server.

Check LabView Help for the correct syntax to use when setting this up.
For example "* .site.com" will allow access to browser addresses ending in "site.com"